

Efficient numerical integration of neutrino oscillations in matter

F. Casas

*Departament de Matemàtiques and IMAC, Universitat Jaume I, E-12071 Castellón, Spain**

J. C. D'Olivo

*Instituto de Ciencias Nucleares, Universidad Nacional Autónoma de México,
Apartado Postal 70-543, 04510 México, D.F., Mexico†*

J. A. Oteo

Departamento de Física Teórica, Universidad de Valencia, 46100-Burjassot, Valencia, Spain‡
(Dated: November 22, 2016)

A special purpose solver, based on the Magnus expansion, well suited for the integration of the linear three neutrino oscillations equations in matter is proposed. The computations are speeded up to two orders of magnitude with respect to a general numerical integrator, a fact that could smooth the way for massive numerical integration concomitant with experimental data analyses. Detailed illustrations about numerical procedure and computer time costs are provided.

I. INTRODUCTION

Discovered with atmospheric [1] and solar neutrinos [2], neutrino oscillations [3, 4] have been corroborated by experiments using neutrinos from nuclear reactors and accelerators [5]. These observations have established beyond doubt that neutrinos have masses and mix [6], which has a fundamental impact not only on particle physics, but also on astrophysics and cosmology. For three flavors (ν_e, ν_μ, ν_τ) the oscillation frequencies in vacuum are characterized by two independent differences between the squared masses: $\delta m^2 \equiv m_2^2 - m_1^2$ (solar) and $\Delta m^2 \equiv m_3^2 - (m_1^2 + m_2^2)/2$ (atmospheric), with $\Delta m^2 \gg \delta m^2$. While propagating through ordinary matter, the coherent forward scattering of ν_e on electrons and nucleons differ from those of ν_μ and ν_τ . As a consequence, the oscillation probabilities are modified in a non trivial manner via the Mikheyev-Smirnov-Wolfenstein (MSW) mechanism [7]. Matter effects are specially significant in the Sun and other astrophysical objects and events, in particular, in core-collapse supernovae [8].

Out of the six oscillation parameters, δm^2 , $|\Delta m^2|$, and the mixing angles $\theta_{12}, \theta_{23}, \theta_{13}$, are known at present [9]. Thanks to the MSW effect involved in the flavor transformations of solar neutrinos we know that $\delta m^2 > 0$ (i.e., $m_2 > m_1$). The sign of Δm^2 depends on the ordering of the mass spectrum, positive for normal hierarchy ($m_3 > m_{1,2}$) and negative for inverted hierarchy ($m_3 < m_{1,2}$). The data at hand neither allow to establish the ordering of the neutrino masses nor the value of the phase δ associated with a possible CP-violation in the leptonic sector. We also ignore if neutrinos are their own anti-particles. An intense research program is underway to address these and other important questions regarding

neutrino physics [10].

Cosmic neutrinos offer opportunities both to understand the properties and behavior of neutrinos, as to probe the sources that produce them. Until now, two astrophysical objects have been observed with neutrinos, the Sun and the supernova SN1987A. High-energy extraterrestrial neutrinos have been observed by IceCube but their origin is still unclear [11]. The detection of solar neutrinos not only confirmed that nuclear fusion reactions power the Sun, but also solved the solar neutrino puzzle, providing the only observed matter effect on the neutrino propagation to date [12]. On the other hand, observing supernova (SN) neutrinos, both Galactic and relic, is in the agenda of the future large underground detectors [13]. As they stream out of a SN, in addition to the influence of the stellar matter (through the MSW effect), neutrinos are subject to the interaction with other neutrinos and antineutrinos [14]. The latter effect turns flavor evolution into a highly non-linear problem [15] and gives rise to collective oscillations of the neutrino gas (see [16] for recent reviews and extensive lists of references). This is a complex and rich phenomenon and, despite the significant progress made, a complete picture of it is still lacking. Limitations of the previous studies have been recently recognized and open issues pointed out [17]. More realistic numerical studies with a full three-flavor mixing seem necessary to make further progress.

In a medium, the evolution equation of the neutrino flavor amplitudes is not (in general) analytically solvable. In the case of ordinary matter, the way from collected data to sound determination of the values of the parameters requires massive numerical integrations of a linear, homogeneous system of ordinary differential equations (ODE) with coefficients depending on the distance neutrinos travel along a medium. However, there is a lack of discussion in the literature about the numerical methods used to this end. The efficiency of the procedure, i.e., the interplay between numerical accuracy and computation time, seems to be given for granted. Even though it is understandable, since it does not constitute an essential

*Electronic address: casas@mat.uji.es

†Electronic address: dolivo@nucleares.unam.mx

‡Electronic address: oteo@uv.es

part of the research, is noteworthy the very few references existing in this regard in the specialized literature. To our knowledge, only two papers have addressed this issue, both of them within the realm of neutrino astrophysics [18, 19]. Some articles in the same field have dealt with questions such as convergence and stability of the numerical solutions found by using ODE solvers [20], but no attention was paid to alternative algorithms.

The integration of the differential equations governing the matter neutrino oscillations with classical general purpose methods of numerical integration (e.g. Runge-Kutta schemes) does not preserve the norm of the solution at every integration step. The value of the norm becomes rather a test of the quality of the integration itself. In contrast, the so-called geometric numerical integrators are not flawed by such an issue, since they are designed to preserve this property exactly (up to round-off error). They have been developed in recent years within the area of numerical analysis of differential equations [21–23] with the aim of preserving under discretization qualitative (very often geometric) properties the differential equation has. It is the main purpose of this work to argue in favor of their use as a convenient alternative to carry out neutrino oscillation numerical integrations. Many geometric integrators exist but, to be concrete, we will focus on a particular class based on the Magnus expansion (ME) [24–26]. We explain shortly the way the algorithms are built up and give details that facilitate their coding. Other similar procedures could be applied with the same purpose without further complications.

ME provides an exponential representation for the solution of linear, homogeneous systems of differential equations. Originally considered as a procedure to build approximate analytical solutions of the time-evolution operator, in the sense of Perturbation Theory, its application as an efficient numerical integrator constitutes a relatively recent development [21, 25]. Years ago, two of us introduced the use of the time-evolution operator to study matter neutrino oscillations and applied its representation in terms of the ME to incorporate non-adiabatic effects into the flavor transitions [27, 28]. A distinctive feature of the ME is that the approximate solution provided by the procedure shares with the exact solution relevant geometric properties. More specifically, for the case under consideration, the relevant property corresponds to the unitary character of the time-evolution operator and, consequently, the norm of the solution vector. In a numerical calculation, this is verified at every single integration step and, therefore, probability is conserved by the final solution obtained through the composition of the successive steps. As already mentioned, this characteristic is not shared by classical numerical integrators like, for instance, those of the Runge-Kutta class. Another, more practical, advantage of a ME based method is the use of larger values of the integration step which conveys shorter CPU time to the same solution accuracy.

The paper is organized as follows. In Section II we write down the neutrino flavor evolution equation to be

integrated. The basics of the ME are given in Section III, and their implementation in the problem at hand is carried out in Section IV. Section V contains the outputs of the integration for three neutrino oscillations in the Sun and a supernova, modeled by simple density profiles. A comparison in terms of the computing efficiency is also provided for an algorithm based on ME and a standard integration routine. The discussion of the results is in Section VI.

II. THE FLAVOR EVOLUTION EQUATION

Except for some anomalous results from short baseline measurements which require further clarification [29], existing data can be interpreted with the simplest extension of the standard model to incorporate nonzero masses of mixed active neutrinos. Namely, within a framework where the three known flavor states $|\nu_\alpha\rangle$ ($\alpha = e, \mu, \tau$) are linear combinations of the states $|\nu_i\rangle$ with masses m_i ($i = 1, 2, 3$):

$$|\nu_\alpha\rangle = \sum_i U_{\alpha i}^* |\nu_i\rangle. \quad (1)$$

The coefficients $U_{\alpha i}$ are elements of the unitary mixing matrix U that appears in the charged current, the so called Pontecorvo–Maki–Nakagawa–Sakata (PMNS) matrix. For Dirac neutrinos this matrix is customarily expressed as

$$U = \mathcal{O}_{23}\Gamma\mathcal{O}_{13}\Gamma^\dagger\mathcal{O}_{12}, \quad (2)$$

where the orthogonal matrices \mathcal{O}_{ij} represent rotations by angles $\theta_{ij} \in [0, \pi/2]$ in the respective planes, while $\Gamma = \text{diag}(1, 1, e^{i\delta})$, with $\delta \in [0, 2\pi]$. If neutrinos are Majorana particles U has to be multiplied by the right by another diagonal matrix $\Gamma_M = \text{diag}(1, e^{i\delta_1}, e^{i\delta_2})$. The two additional physical phases do not play a role in neutrino oscillations and are therefore omitted in the analysis of the phenomenon [30].

Let us consider a neutrino ν_α produced within a medium at time t_0 . The state of the system $|\psi(t)\rangle$ at time $t \geq t_0$ can be expressed as $|\psi(t)\rangle = \sum_\beta \psi_\beta(t) |\nu_\beta\rangle$, with $\psi_\beta(t_0) = \delta_{\alpha\beta}$. The probability to have a state of flavor β at a point $r = t$ ($\hbar = c = 1$) is

$$P_{\alpha\beta}(r) = |\psi_\beta(r)|^2. \quad (3)$$

Once the neutrinos leave the medium the amplitudes evolve according to the equation that governs vacuum oscillations, whose solution is simpler when written in the basis of the mass eigenstates. According to Eq. (1), denoting by $\mathcal{A}_j = \phi_j(r_\star)$ the probability amplitude of having a ν_j at the edge of the medium, for $r \geq r_\star$ we can write

$$\psi_\beta(r) = \sum_{j=1}^3 U_{\beta j} \mathcal{A}_j \exp(-iE_j L), \quad (4)$$

where $E_j = \sqrt{|\mathbf{p}|^2 + m_j^2}$ and $L = r - r_*$ is the distance travelled by the neutrinos in vacuum. Substituting (4) into Eq. (3) we obtain

$$P_{\alpha\beta} = \sum_j |U_{\beta j}|^2 |\mathcal{A}_j|^2 + 2 \sum_{i>j} \text{Re} [U_{\beta i} U_{\beta j}^* \mathcal{A}_i \mathcal{A}_j^* \exp(-i\Delta_{ij}L)]. \quad (5)$$

The quantity $\Delta_{ij} = \Delta m_{ij}^2/2E$, with $E = |\mathbf{p}|$, is the oscillation wave number associated with the squared mass difference $\Delta m_{ij}^2 = m_i^2 - m_j^2$.

As a result, the problem of calculating $P_{\alpha\beta}$ reduces to determine the quantities \mathcal{A}_j , i.e., find the mass eigenstate amplitudes $\phi_j(r)$ within the medium, subject to the initial condition $\phi_j(r_0) = U_{\alpha j}^*$. For relativistic neutrinos propagating in normal matter, after subtracting a global phase, the evolution equation for these amplitudes has the form

$$i \frac{d}{dr} \Phi(r) = [H_0 + v(r)U^\dagger V U] \Phi(r), \quad (6)$$

with $\Phi^T(r) = (\phi_1(r), \phi_2(r), \phi_3(r))$, U the mixing matrix, and $V = \text{diag}(1, 0, 0)$. The first term between brackets is the Hamiltonian matrix that governs the flavor evolution in vacuum $H_0 = \text{diag}(0, \Delta_{21}, \Delta_{31})$, while the second term accounts for the matter effects due to the coherent interaction of the neutrinos with the background particles. The quantity

$$v(r) = \sqrt{2} G_F n_e(r) \quad (7)$$

denotes the difference between the effective potential energies of the ν_e and the $\nu_{\mu,\tau}$. Here, G_F is the Fermi constant and $n_e(r)$ is the number density of electrons along the neutrino path.

If neutrinos propagate only in vacuum, then $v(r) = 0$ along their entire way to the detector and L corresponds to the distance from the production point at r_0 . Replacing \mathcal{A}_j by $U_{\alpha j}^*$ in Eq. (5) we recover the usual formulas for the oscillation probabilities in vacuum.

Since \mathcal{O}_{12} and Γ commute, the PMNS matrix can be written as $U = \mathcal{O}_{23}\Gamma\mathcal{O}\Gamma^\dagger$, where $\mathcal{O} = \mathcal{O}_{13}\mathcal{O}_{12}$. From this, taking into account that the commutators $[V, \mathcal{O}_{23}]$, $[V, \Gamma]$, and $[H_0, \Gamma]$ vanish, we obtain

$$i \frac{d}{dr} \Psi(r) = H(r) \Psi(r), \quad (8)$$

with $\Psi(r) = \Gamma^\dagger \Phi(r)$ and

$$H(r) = H_0 + v(r)W. \quad (9)$$

The matrix W is given by $W = \mathcal{O}^T V \mathcal{O}$, where \mathcal{O}^T is the transpose of the orthogonal matrix \mathcal{O} . Explicitly,

$$W = \begin{pmatrix} c_{13}^2 c_{12}^2 & c_{12} s_{12} c_{13}^2 & c_{12} c_{13} s_{13} \\ c_{12} s_{12} c_{13}^2 & s_{12}^2 c_{13}^2 & s_{12} c_{13} s_{13} \\ c_{12} s_{13} c_{13} & s_{12} c_{13} s_{13} & s_{13}^2 \end{pmatrix}, \quad (10)$$

with $s_{ij} = \sin \theta_{ij}$ and $c_{ij} = \cos \theta_{ij}$. Note that $H(r)$ is a real and symmetric matrix that can be diagonalized by an r -dependent orthogonal transformation. It does not contain θ_{23} nor δ and then its eigenvalues, which are identical to those of the Hamiltonian in Eq. (6), do not depend on these two parameters.

Typically, the oscillatory interference terms in (5) average to zero for neutrinos traveling a long distance to the Earth. Under such circumstances, the average probability of finding a ν_β at a detector in the Earth is given by the incoherent superposition

$$\langle P_{\alpha\beta} \rangle = \sum_j |U_{\beta j}|^2 \mathcal{P}_j, \quad (11)$$

where $\mathcal{P}_j \equiv |\mathcal{A}_j|^2$ denotes the probability to have a mass eigenstate at the surface of the star. Neutrinos that travel a certain distance through the Earth before reaching the detector will experience oscillations in the terrestrial matter. As a consequence, $U_{\beta i}$ in $\langle P_{\alpha\beta} \rangle$ has to be replaced by the amplitude for a ν_i emerging from the Earth be in a flavor state ν_β . We do not consider this effect here but, whenever needed, it can be easily incorporated in our approach.

If the initial state corresponds to an electron neutrino, then

$$\Psi(r_0) = \begin{pmatrix} c_{12} c_{13} \\ s_{12} c_{13} \\ s_{13} \end{pmatrix} \quad (12)$$

and, according to Eq. (11), the average survival probability reads

$$\langle P_{ee} \rangle = c_{12}^2 c_{13}^2 \mathcal{P}_1 + s_{12}^2 c_{13}^2 \mathcal{P}_2 + s_{13}^2 \mathcal{P}_3, \quad (13)$$

where

$$\sum_{j=1}^3 \mathcal{P}_j = 1 \quad (14)$$

by conservation of probability. For solar neutrinos the flux of electron neutrinos at the detection point F_{ν_e} is obtained multiplying the original flux $F_{\nu_e}^{(0)}$ by the expression of $\langle P_{ee} \rangle$ as a function of energy.

On the other hand, neutrinos (and antineutrinos) of all species are produced during a core-collapse SN. The flux of ν_e arriving at Earth may be written as [31]

$$F_{\nu_e} = F_{\nu_e}^0 \langle P_{ee} \rangle + F_{\nu_x}^0 (1 - \langle P_{ee} \rangle), \quad (15)$$

where ν_x stands for either ν_μ or ν_τ . In the deepest regions the neutrino density is now so high that neutrino-neutrino interactions are dominant and produce novel collective effects [16]. The net result is the modification of the primary fluxes $F_{\nu_\alpha}^0$ because of spectral swaps in certain energy intervals. Usually, this happens within the first few 100 km, much before neutrinos experience the MSW conversions we consider here, which can be incorporated separately.

In the rest of the article, we will focus on the determination of the vector $\Psi(r)$ or equivalently the quantities \mathcal{P}_j calculated numerically by means of the Magnus procedure. When substituted into Eq. (13) they render us $\langle P_{ee} \rangle$ as a function of the energy. The corresponding curves for the Sun and a SN are plotted in Figure 1. They were obtained by means of the method denoted as M4 in Section IV and using a set of the oscillation parameters consistent with the experimental results.

III. MAGNUS EXPANSION: BASICS

For the sake of completeness we present here the basics of ME [26]. The explanations are oriented toward its use as a numerical integrator.

Given the matrix differential equation

$$\frac{d}{dt}Y(t) = A(t)Y(t), \quad A \in \mathbb{C}^{n \times n}, \quad Y \in \mathbb{C}^n, \quad (16)$$

with the initial condition $Y(t_0) = Y_0$, the Magnus approach consists in looking for an exponential representation of the solution

$$Y(t) = \exp\{\Omega(t, t_0)\}Y_0, \quad (17)$$

with

$$\Omega(t, t_0) \in \mathbb{C}^{n \times n}, \quad \Omega(t_0, t_0) = 0. \quad (18)$$

When the matrix elements of A are t -independent, then $\Omega(t, t_0) = (t - t_0)A$. In general, $\Omega(t, t_0)$ is given as a series expansion

$$\Omega(t, t_0) = \sum_{k=1}^{\infty} \Omega_k(t, t_0), \quad \Omega_k(t_0, t_0) = 0, \quad (19)$$

where the term Ω_k is built up from multiple integrals of nested commutators of k matrices A evaluated at k different times [24]. The complexity of the individual terms largely increases with the index k . In particular, the first three ones read

$$\Omega_1(t, t_0) = \int_{t_0}^t dt_1 A_1, \quad (20)$$

$$\Omega_2(t, t_0) = \frac{1}{2} \int_{t_0}^t dt_1 \int_{t_0}^{t_1} dt_2 [A_1, A_2], \quad (21)$$

$$\Omega_3(t, t_0) = \frac{1}{6} \int_{t_0}^t dt_1 \int_{t_0}^{t_1} dt_2 \int_{t_0}^{t_2} dt_3 \{ [A_1, [A_2, A_3]] + [[A_1, A_2], A_3] \}, \quad (22)$$

where $A_i \equiv A(t_i)$ and the square brackets stand for the commutator: $[A, B] \equiv AB - BA$, for A, B , matrices of appropriate dimensions.

For the purposes of using ME as a numerical integrator that furnishes a solution $Y(t_f)$ starting from Y_0 , the question focuses on how to handle efficiently a single

integration step. Namely, one considers certain grid points $t_0 < t_1 < \dots < t_N = t_f$ of the time interval $[t_0, t_f]$ with associated time increments $h_n = t_{n+1} - t_n$ for $0 \leq n \leq N-1$ and then determines

$$Y(t_{n+1}) = \exp\{\Omega(t_n + h_n, t_n)\}Y(t_n). \quad (23)$$

After that, iteration yields the solution in N steps

$$Y(t_f) = \prod_{n=0}^{N-1} \exp\{\Omega(t_n; h_n)\}Y_0, \quad (24)$$

with $\Omega(t_n; h_n) \equiv \Omega(t_n + h_n, t_n)$. When the matrix A is anti-Hermitian then the matrix $\prod_{n=0}^{N-1} \exp\{\Omega(t_n; h_n)\}$ constitutes a unitary approximation to the time-evolution operator between t_0 and t_f , since it is the product of N unitary matrices.

The procedure has to face three problems of different nature:

- Truncate the series (19) for $\Omega(t_n; h)$ at an appropriate index,

$$\Omega^{(p)}(t_n; h_n) = \sum_{k=1}^p \Omega_k(t_n; h_n). \quad (25)$$

- Generate the quadratures to approximate the multivariate integrals in the truncated series.
- Compute the exponential of the resulting square matrix.

The last point will be treated later on and an explicit algebraic form given for the three neutrino case. The other two are solved at once and depend on the order of approximation of the method. This is defined as the order r of the Taylor expansion that gives $Y(t_n + h_n)$, up to $O(h_n^{r+1})$, out of the exact value $Y(t_n)$. One important point is that the order of the quadratures required in the numerical approximation of Ω_k in the truncated series (25) is not greater than the order of approximation r . This fact is exploited so as to minimize the number of operations in the algorithm. Moreover, since ME is time symmetric, for achieving an integration method of order $2r$ (with $r > 1$) only terms up to $p = 2r - 2$ are required in the series expansion (19). Book-keeping technicalities may be found in the literature [21, 25].

IV. NUMERICAL INTEGRATION

The differential system that rules the evolution of the three-neutrino amplitudes in matter (Eq. (8)) is of the type in (16) with $\mathbf{n} = 3$, $t = r$, and the identifications

$$A \equiv -iH, \quad Y \equiv \Psi. \quad (26)$$

Now, we apply the approach outlined in the previous section to solve numerically such equation with the initial

condition (12). Since each term in the series expansion of $\Omega(t, t_0)$ is anti-Hermitian, ME provides a representation of $\exp\{\Omega(t, t_0)\}$ in (17) that is unitary by construction, even if the series is truncated. This property is preserved by the factorized expression given in Eq. (24) and by the algorithm used to compute each factor. As a consequence, condition (14) is fulfilled automatically by the \mathcal{P}_j calculated using the procedure.

We assume normal hierarchy and adopt the best-fit values of the 3-neutrino oscillation parameters, derived from a global fit of the current neutrino data [9]: $\Delta m_{21}^2 = 7.54 \times 10^{-5} \text{eV}^2$, $\Delta m_{31}^2 = 2.47 \times 10^{-3} \text{eV}^2$, $\sin^2 \theta_{12} = 0.308$, $\sin^2 \theta_{23} = 0.437$, and $\sin^2 \theta_{13} = 0.0234$. In addition, the distance is expressed in units of the solar radius $R_\odot = 6.96 \times 10^5 \text{ km}$. Thus, written in terms of the dimensionless variable $\xi \equiv r/R_\odot$, Eq. (8) becomes

$$i \frac{d\Psi}{d\xi} = [H_0 + v(\xi)W] \Psi, \quad (27)$$

with W the matrix given in (10) and

$$H_0 = \frac{a}{\mathcal{E}} \begin{pmatrix} 0 & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (28)$$

Here \mathcal{E} is the numerical value of the neutrino energy in MeV and $a = 4.35196 \times 10^6$ and $b = 0.030554$ are dimensionless parameters.

The conclusions about the numerical efficiency are essentially independent on the initial integration point. To be concrete, for the computations we use $\xi_0 = 0.1$ in the case of the Sun and $\xi_0 = 0.02$ in the case of a SN. We also take the supernova's radius equal to $20 R_\odot$. Accordingly, the integration yields $\Psi(1)$ and $\Psi(20)$ for the Sun and a SN, respectively. The integration procedure requires a particular shape of the function $v(\xi)$ to be provided. In order to keep as clear as possible the technical explanations we consider two simple profiles of the electron number density existing in the literature. For the Sun, we model the density by an exponential profile $v(\xi) = \gamma \exp(-\eta\xi)$, with $\gamma = 6.5956 \times 10^4$ and $\eta = 10.54$ [32]. In the SN case, we adopt the power law used in [33], namely, $v(\xi) = \gamma/\xi^3$, with $\gamma = 52.934$. Let us remark that $v(\xi)$ does not need to be analytically defined, nor to be a continuous function, nor to have continuous derivatives. This applies, for instance, to neutrinos that traverse the Earth, which radial matter density is represented by a discontinuous piecewise function according to the Preliminary Reference Earth Model (PREM) [34].

Next, we describe the main features of the numerical integration of Eq. (8) by means of two particular schemes based on the ME of orders 2 and 4. Starting the integration at ξ_0 , with the initial value $\Psi(\xi_0)$ we build up the approximate solution at $\Psi(\xi_1 = \xi_0 + h_1)$. At a generic point ξ_n , the solution $\Psi(\xi_n)$ determines $\Psi(\xi_n + h_n)$ as

$$\Psi(\xi_{n+1}) = \exp\left\{\Omega^{[r]}(\xi_n; h_n)\right\} \Psi(\xi_n). \quad (29)$$

Here $\Omega^{[r]}$ denotes an approximation to the truncated Magnus series (25) of order r in h_n , with $r = 2, 4$. The iteration stops after N iterations at $\xi = 1, 20$.

A. Order 2 formula: M2

The method of order $r = 2$ is particularly simple for two reasons. First, just the term (20), namely,

$$\Omega_1(\xi_n; h_n) = -i \int_{\xi_n}^{\xi_n + h_n} dt H(t), \quad (30)$$

has to be taken into account in the series (25). Second, the quadrature for this integral requires only one evaluation point to obtain a second order approximation. Thus, for the exponent in (29) we get

$$\Omega^{[2]}(\xi_n; h_n) = -iH(\bar{\xi})h_n = -i(H_0 + \bar{v}W)h_n, \quad (31)$$

with $\bar{\xi} \equiv \xi_n + h_n/2$. The quantity $\bar{v} \equiv v(\bar{\xi})$ must be re-evaluated at every step. This scheme is also known as the exponential midpoint rule and we will refer to it as M2.

B. Order 4 formula: M4

With $p = 2$ in the truncated series (25) one ends up with a method of order $r = 4$ if the corresponding integrals (20) and (21) are approximated by a 2-point Gauss-Legendre quadrature rule [21, 25]. Specifically, given the quadrature points

$$\xi_\pm = \xi_n + \left(1 \pm \frac{1}{\sqrt{3}}\right) \frac{h_n}{2}, \quad (32)$$

with $\xi_- < \xi_+$ and defining the quantities

$$H_\pm = H(\xi_\pm), \quad (33)$$

the one-step integration exponent of order 4 can be cast into the form

$$\Omega^{[4]}(\xi_n; h_n) = -i(H_+ + H_-) \frac{h_n}{2} + \frac{\sqrt{3}}{12} [H_-, H_+] h_n^2, \quad (34)$$

where, as before, the square brackets stand for the matrix commutator. Alternatively, the Simpson quadrature rule also yields an equivalent fourth-order approximation [25].

Working out equation (34) for the matrix H given in (9), we arrive at

$$\begin{aligned} \Omega^{[4]}(\xi_n; h_n) = & -i \left(H_0 + \frac{1}{2}(v_+ + v_-)W \right) h_n \\ & + \frac{\sqrt{3}}{12} (v_+ - v_-) [H_0, W] h_n^2, \end{aligned} \quad (35)$$

where $v_\pm \equiv v(\xi_\pm)$, are the only quantities to be re-evaluated at every integration step. The matrix $[H_0, W]$ is built up only once, namely at the beginning of the integration process. This method will be referred to as M4.

C. Variable step size implementation

The easiest way to implement the previous methods is by considering a constant step size, i.e., by taking $h = (\xi_f - \xi_0)/N$ and then setting $\xi_n = \xi_0 + nh$. This implementation tends to be inefficient, however, since the solution $\Psi(\xi)$ may experience rapid changes along the evolution on some intervals and evolve slowly on some others. Thus, it is better to adjust h_n accordingly as the integration proceeds. There are several techniques for doing this automatically in such a way that the local error is below a prescribed tolerance `tol`, one of the most common being the local extrapolation procedure [35, 36]. In our setting this approach can be summarized as follows.

We produce two numerical solutions at ξ_{n+1} according with M2 and M4 above,

$$\hat{\Psi}_{n+1} = e^{\Omega^{[2]}(\xi_n, h_n)} \Psi_n, \quad \Psi_{n+1} = e^{\Omega^{[4]}(\xi_n, h_n)} \Psi_n, \quad (36)$$

respectively. Then the quantity

$$E_r = \|\hat{\Psi}_{n+1} - \Psi_{n+1}\| \quad (37)$$

can be used to estimate the local error corresponding to M2. If E_r computed at ξ_{n+1} is below `tol`, then the step from ξ_n to ξ_{n+1} is accepted and then we proceed to compute the approximation to the solution at ξ_{n+2} . If $E_r > \text{tol}$, then the approximation at ξ_{n+1} is rejected and a smaller step is chosen to compute a new approximation at ξ_{n+1} . In either case, the value of the new step is given by [36]

$$h_{\text{new}} = s h_c \left(\frac{\text{tol}}{E_r} \right)^{1/3}, \quad (38)$$

where h_c denotes the current value of the step size and s is a “safety factor” chosen to decrease the probability of a rejection at the next step. For our problem a good choice is $s = 0.8$. Notice that once the step $\xi_n \rightarrow \xi_{n+1}$ has been completed, we have two numerical approximations at x_{n+1} : $\hat{\Psi}_{n+1}$ and Ψ_{n+1} , so that it is possible to choose one or the other as the approximation to $\Psi(\xi_{n+1})$. In local extrapolation, one advances with the higher-order result Ψ_{n+1} (hence the name).

Since the most time-consuming part of the Magnus methods corresponds to the computation of the exponential, evaluating directly E_r as (37) may increase considerably the total computational cost of the algorithm. To avoid that we can express

$$\hat{\Psi}_{n+1} - \Psi_{n+1} = \left(e^{\Omega^{[2]}} - e^{\Omega^{[4]}} \right) \Psi_n = (e^Z - I) \Psi_{n+1}, \quad (39)$$

where

$$Z = \log(e^{\Omega^{[2]}} e^{-\Omega^{[4]}}) = \Omega^{[2]} - \Omega^{[4]} - \frac{1}{2}[\Omega^{[2]}, \Omega^{[4]}] + \dots \quad (40)$$

Working out this expression and expanding e^Z , we arrive at

$$E_r = \|(h_n^2 S_1 + h_n^3 S_2 + \frac{1}{2} h_n^4 S_1^2) \Psi_{n+1}\| + \mathcal{O}(h_n^5), \quad (41)$$

where

$$\begin{aligned} S_1 &= -\frac{\sqrt{3}}{12}(v_+ - v_-)[H_0, W] \\ S_2 &= i\frac{\sqrt{3}}{24}(v_+ - v_-)\left([H_0, [H_0, W]] \right. \\ &\quad \left. + \frac{1}{2}(v_+ + v_-)[W, [H_0, W]]\right). \end{aligned} \quad (42)$$

Notice that estimating E_r according to (41) only requires additional combinations of v_{\pm} at each step and the evaluation of the nested commutators $[H_0, [H_0, W]]$ and $[W, [H_0, W]]$ at the beginning of the integration. In practice, we will scale the i -th component of $\hat{\Psi}_{n+1} - \Psi_{n+1}$ by a factor $d_i = |(\Psi_{n+1})_i|$ to work with relative errors. As for the initial step size, a possible choice is just $h_0 = \text{tol}/2$.

More sophisticated schemes exist, of course, for step-size control that are employed in commercial computer packages. The previous method, although simple, is quite efficient and well adapted for the problem at hand.

As with respect to the computation of the matrix exponential, the Appendix below provides an analytic formula for that. It has been written down so as to cope with eventual quasi-degeneracy of eigenvalues (causing extra rounding-off errors) and to allow minimal amount of arithmetic. Its direct coding is straightforward and saves computing time with respect to general purpose routines developed for generic dimension matrices.

V. RESULTS

All the result we will show correspond to outputs of double precision computations carried out in Fortran with no optimization during the compilation. They have been replicated in Matlab albeit investing longer CPU times.

To test the performance of the M4 method we have carried out the very same computations using the well known ODE solver Dopri5(4) designed by Dormand and Prince [37]. It is an adaptive step-size explicit Runge-Kutta method of order five that has embedded a fourth-order approximation used to estimate the error and applies local extrapolation. The Matlab procedure `ode45`, in particular, is based on this integrator.

Here we have used the standard Fortran implementation of Dopri5(4) called DOPRI5 available at [38]. Given a particular level of accuracy, the focus will be on the relatively shorter CPU time needed by M4 in comparison with DOPRI5, rather than on the absolute measures of CPU time. It might be, however, of interest to point out that computations were carried out on an Intel Core Duo E8400 processor running at 3 GHz.

In Figure 1 we plot the average electron-neutrino survival probability (Eq.(13)) at the edge of Sun (dashed line) and a SN (solid line) as a function of the neutrino energy. These curves reproduce results of the same kind available in a number of references. As a test, we mention that the theoretical asymptotic value $\langle P_{ee} \rangle = 0.547829$

is correctly reproduced, both in the high energy and the low energy limits.

Figures 2 and 3 contain the main output of the present work. They show the relationship between the accuracy obtained for a numerical solution and the computational cost required as measured by the CPU time invested to achieve it. Every CPU time corresponds to a particular value of the tolerance `tol`. The smaller `tol`, the larger CPU time. These curves allow one to compare the relative efficiency of the two ODE solvers tested: M4, and DOPRI5. Two different representative values of the neutrino energy have been considered, as specified in each diagram. Given a value for the relative error (to be defined in next paragraph) of the numerical solution of (27) then the corresponding abscissa of every curve determines the CPU time that was invested in the computation from $\xi = 0.1$ up to $\xi = 1$ (Sun) and from $\xi = 0.02$ up to $\xi = 20$ (SN). Certainly, the CPU time is not an absolute measure of efficiency since it depends on the type of processor, the programmer expertise, and the compiler. Nevertheless, the location of the curves in the plots is a reasonable relative measure of efficiency of each method, provided the runs are carried out on the same computer and compiler. This type of plots allow us to visualize the balance between the accuracy required in the solution and the affordable numerical effort.

In order to build up the relative error we employed an “exact” numerical solution $\Psi^{ref}(\xi)$ obtained by means of the command `NDSolve` of *Mathematica* with stringent requirements to guarantee high accuracy. Specifically, the relative error is determined as the norm of the vector with components given by the quantities $(\psi_j(\xi) - \psi_j^{ref}(\xi))/\psi_j^{ref}(\xi)$ ($j = 1, 2, 3$), evaluated at $\xi = 1$ (Sun) or $\xi = 20$ (SN). Notice that massive numerical computations cannot afford the huge amount of CPU time required by the procedure to obtain $\Psi^{ref}(\xi)$. Hence, the need for efficient ODE solvers.

When one plots the error obtained with an integrator of order r as a function of the computational cost in a double logarithmic scale, in the limit $h_n \rightarrow 0$ one obtains a straight line with slope $-r$, so that usually the order of the method can be deduced at once from the diagram. For this reason, we have included the straight line with slope -4 , as visual reference.

Given that \mathcal{E} appears in the denominator of (28) the instantaneous oscillation frequencies of the solution increase roughly as the inverse of the neutrino energy. Thus, it is expected that at low energies the integration should take longer times. The results we report in Figure 2 (solar neutrinos) stand for the relative error in the solution at the edge of the Sun for two energies, $E = 1$ MeV and 10 MeV, and confirm this point. This effect is more dramatic for DOPRI5 than for M4. For fixed energy, M4 performs significantly better than DOPRI5.

Figure 3 shows the results for SN neutrinos for $E = 15$ MeV and 100 MeV. The dependence of the cost as a function of E follows the same pattern as with solar neutrinos. It is worth noticing the remarkable improvement achieved

with the Magnus method in all cases in comparison to the standard integration procedure.

VI. DISCUSSION

For neutrinos propagating in a medium, numerical calculation of the flavor amplitudes is very often a costly process due to the highly oscillatory character of the solutions of the Schrödinger-like equation that governs the problem. Given the interest in analyzing the neutrino flavor evolution in a variety of settings, any novel procedure able to provide accurate approximations with a reduced computation time is of undeniable benefit. The purpose of this article is precisely to present one of such techniques, namely an explicit fourth-order integrator based on the ME for linear differential systems, that is equipped with an adaptive step-size strategy to render it more efficient in practice.

To evaluate the performance of the new integrator M4 we compared it with the well known routine DOPRI5 based on an explicit 5th-order Runge-Kutta method with variable step size. The comparison was made for three-mixed neutrinos evolving in the Sun and a type-II SN. We use realistic values for the oscillations parameters (mixing angles and square-mass differences). For simplicity, we modeled the matter densities in terms of simple analytical functions, but the procedure can be easily adapted to more complex situations, like a piecewise profile or one defined numerically. In our numerical tests, we found that results with the same accuracy for the ν_e survival probability were produced by M4 with a computational cost between one and two orders of magnitude lower than DOPRI5, depending on the value of the neutrino energy.

The applicability of the procedure is not constrained by the dimension of the system of linear differential equations. Therefore, it can be used to examine matter neutrino oscillations in the presence of one or more *sterile neutrinos*, i.e., massive singlet leptons that do not interact with the weak gauge bosons and mix with the standard active neutrinos. As a result of the mixture of the active and sterile neutrinos, the PMNS matrix becomes a nonunitary part of a larger mixing matrix. When looking for possible effects of such non-unitarity in matter neutrino oscillations it might be advisable to employ numerical methods that are not affected by any intrinsic violation of unitarity. As already emphasised, this is precisely an additional advantage of the Magnus integrator, which, by construction, will preserve the norm of the solution of the whole system, irrespective of the number of mixed neutrinos.

As has been formulated, our procedure cannot be used to integrate nonlinear differential equations like those arising in a SN core. Nevertheless, a conveniently modified version might still be applied in these cases [39].

In Ref. [19], it has been presented a detailed proposal, based on a Monte Carlo method, for the (not preserving unitarity) numerical integration of the evolution opera-

tor for the two neutrino system. Although the aim of that work was not oriented to a systematic study of the CPU cost, the method seems to be quite expensive. The technique might be useful in long profiles, such as the core-collapse SN, but would not be the best choice in simpler situations, like the short, solar profile.

In the case of two neutrino generations, a nonlinear transformation of the linear differential system has been proposed as an alternative [19]. In terms of hyperspherical polar coordinates, a new set of four nonlinear, real-valued equations is obtained. One of them, which stands for the norm of the solution, becomes a constant of motion and hence the norm is preserved, no matter the quality of the integration. The numerical integration proceeds actually with a system of three equations for which a preliminary analysis of singularities is needed. An appropriate generalization of this scheme to three generations could be a further alternative starting point. We have not yet explored this issue.

In Perturbation Theory, the use of the interaction picture and the adiabatic basis are common tools. As far as the numerical integration of Eq. (27) is concerned, a change of picture transforms the slow varying coefficients system into a linear differential system whose coefficients oscillate rapidly, which actually is a harder problem. Anyhow, we carried out an implementation of such an approach [40] and convinced ourselves that any eventual advantage in the integration process does not compensate the extra algebra needed. For this reason, we have not included it here.

Interrelation between Numerical Analysis and Physics has proved to be of mutual benefit. We hope that this work will help to motivate neutrino physics researchers to incorporate geometric ODE solvers as useful tools to do detailed numerical calculations on matter neutrino oscillations.

Acknowledgments

One of the authors (JCD) would like to thank the Departamento de Física Teórica, Universidad de Valencia, where most of his work was done, for its hospitality and its support via an EPLANET grant. This work was supported in part by the Spanish MINECO under grants AYA2013-48623-C2-2 (JAO) and MTM2013-46553-C3 (FC), by DGAPA-UNAM grant PAPIIT IN112213 and PASPA fellowship, and by CONACYT (México) grant 240666.

Appendix: Computation of the matrix exponential

Given a Hermitian matrix A of dimension three, we provide here an efficient algebraic expression to explicitly

compute the unitary matrix $\exp(itA)$, where t is a parameter. For caveats on the numerical computation of the exponential of a matrix see the classical paper by Moler and Van Loan [41].

Without loss of generality we will assume that A is traceless, $\text{Tr}(A) = 0$. Were it not the case, then we can always factorize the problem: $\exp(itA) = \exp(itzI) \exp(itA_0)$, where I stands for the identity matrix and $z = \text{Tr}(A)/3$, $A_0 = A - zI$. Thus, the essential problem reduces to deal with a traceless matrix A_0 . In practice, this conveys a number of important algebraic simplifications.

The eigenvalues of the (traceless) matrix A_0 are all real valued. They are given by the roots of the characteristic equation

$$\lambda^3 - \frac{1}{2}\text{Tr}(A_0^2)\lambda + \text{Det}(A_0) = 0. \quad (\text{A.1})$$

The explicit solutions may then be written down as

$$\lambda_k = \pm 2\sqrt{\frac{p}{3}} \cos \left[\frac{1}{3} \arccos \left(\frac{3q}{2p} \sqrt{\frac{p}{3}} \right) - \frac{2\pi k}{3} \right], \quad (\text{A.2})$$

for $(k = 0, 1, 2)$, where $p \equiv \text{Tr}(A_0^2)/2$, and $q \equiv \text{Det}(A_0)$. The positive sign corresponds to $q \leq 0$, otherwise the negative sign applies. The fact $p > 0$ ensures the real character of the roots.

Finally, using Putzer's algorithm [42] one gets

$$\begin{aligned} \exp(itA_0) = & \exp(i\lambda_0 t) [(1 - \lambda_0(r_0 - \lambda_1 r_1))I \\ & + (r_0 + \lambda_2 r_1)A_0 + r_1 A_0^2], \end{aligned} \quad (\text{A.3})$$

where the eigenvalues have been relabeled such that now $\lambda_0 < \lambda_1 < \lambda_2$, and

$$\begin{aligned} r_0 &= -\frac{1 - \exp(iat)}{a}, \\ r_1 &= -\frac{1}{a-b} \left(\frac{1 - \exp(iat)}{a} - \frac{1 - \exp(ibt)}{b} \right), \end{aligned} \quad (\text{A.4})$$

with $a = \lambda_1 - \lambda_0$, $b = \lambda_2 - \lambda_0$. The formulae (A.4) for the complex coefficients r_0, r_1 , cope with eventual situations where $a, b \ll 1$.

The most expensive part of the computation comes from the matrix product A_0^2 . Thus, considerable amount of CPU time is saved if the explicit numerical matrix computation A_0^2 in (A.3) is avoided and, instead, only the matrix-vector product $A_0\Psi(t)$ and $A_0(A_0\Psi(t))$ are evaluated on the solution vector Ψ along the integration. Notice that in (A.1) the computation of the whole matrix A_0^2 can also be avoided.

-
- [1] Y. Fukuda *et al.* [Super-Kamiokande Collaboration], Phys. Rev. Lett. **81**, 1562 (1998); T. Kajita, Rev. Mod. Phys. **88**, 030501 (2016).
- [2] Q. R. Ahmad *et al.* [SNO Collaboration], Phys. Rev. Lett. **89**, 011301 (2002); A. B. McDonald, Rev. Mod. Phys. **88**, 030502 (2016).
- [3] B. Pontecorvo, Zh. Eksp. Theo. Fiz. **34**, 247 (1957) [Sov. Phys. JETP **7**, 172 (1958)]; B. Pontecorvo, Zh. Eksp. Teor. Fiz. **53**, 1717 (1967) [Sov. Phys. JETP **26**, 984 (1968)]; V. Gribov and B. Pontecorvo, Phys. Lett. B **28**, 493 (1969).
- [4] Z. Maki, M. Nakagawa, and S. Sakata, Prog. Theor. Phys. **28**, 870 (1962).
- [5] M. Zito, Nucl. Part. Phys. Proc. **273-275**, 92 (2016).
- [6] K. Nakamura and S. T. Petcov, *Neutrino Mass, Mixing, and Oscillations*, Review of Particle Physics, Particle Data Group Collaboration 2016 (C. Patrignani *et al.*), Chin. Phys. C **40**, 100001 (2016).
- [7] L. Wolfenstein, Phys. Rev. D **17**, 2369 (1978); S. P. Mikheyev, A. Yu. Smirnov, Yad. Fiz. **42**, 1441 (1985) [Sov. J. Nucl. Phys. **42**, 913 (1985)].
- [8] C. Volpe, J. Phys. Conf. Ser. **631**, 012048 (2015).
- [9] M. C. Gonzalez-García, M. Maltoni, and T. Schwetz, Nucl. Phys. B **908**, 199 (2016); F. Capozzi, G. L. Fogli, E. Lisi, A. Marrone, D. Montanino, and A. Palazzo, Phys. Rev. D **89**, 093018 (2014).
- [10] B. Caccianiga, PoS EPS-HEP2015, 025 (2015).
- [11] M. G. Aartsen *et al.* [IceCube Collaboration], Phys. Rev. Lett. **113**, 101101 (2014).
- [12] For reviews of the current status and outstanding questions on solar neutrinos, see: M. Maltoni and A. Y. Smirnov, Eur. Phys. J. A **52**, 87 (2016); G. D. Orebi Gann, AIP Conf. Proc. **1666**, 090003 (2015).
- [13] F. An *et al.* [JUNO Collaboration], J. Phys. G **43**, 030401 (2016); I. Gil-Botella, *Detection of Supernova Neutrinos*, arXiv:1605.02204 [physics.ins-det].
- [14] G. M. Fuller, R. W. Mayle, J. R. Wilson, and D. N. Schramm, Astrophys. J. **322**, 795 (1987); D. Nötzold and G. Raffelt, Nucl. Phys. B **307**, 924 (1988); J. C. D’Olivo, J. F. Nieves, and M. Torres, Phys. Rev. D **46**, 1172 (1992).
- [15] J. T. Pantaleone, Phys. Lett. B **287**, 128 (1992); Phys. Rev. D **46**, 510 (1992); V. A. Kostelecky and S. Samuel, *ibid.* **49**, 1740 (1994); J. C. D’Olivo and J. F. Nieves, Nucl. Phys. B **35**, 466 (1994); Int. J. Mod. Phys. A **11**, 141 (1996).
- [16] A. Mirizzi *et al.*, Riv. Nuovo Cim. **39**, 1 (2016); H. Duan, G. M. Fuller and Y. -Z. Qian, Ann. Rev. Nucl. Part. Sci. **60**, 569 (2010).
- [17] S. Chakraborty, R. Hansen, I. Izaguirre and G. Raffelt, Nucl. Phys. B **908**, 366 (2016).
- [18] H. Duan, G. M. Fuller, and J. Carlson, Comput. Sci. Dis. **1**, 015007 (2008).
- [19] J. P. Kneller and G. C. McLaughlin, Phys. Rev. D **73**, 05003 (2006).
- [20] S. Sarikas, D. de Sousa Seixas, and G. Raffelt, Phys. Rev. D **86**, 125020 (2012); H. Duan and S. Shalgar, JCAP **10**, 084 (2014); B. D. Keister, Phys. Scripta **90**, 088008 (2015).
- [21] A. Iserles, H. Z. Munthe-Kaas, S. P. Nørsett and A. Zanna, Acta Numerica **9**, 215 (2000).
- [22] E. Hairer, Ch. Lubich and G. Wanner, *Geometric Numerical Integration. Structure-Preserving Algorithms for Ordinary Differential Equations*, 2nd Ed. (Springer, Berlin, 2006).
- [23] S. Blanes and F. Casas, *A Concise Introduction to Geometric Numerical Integration*, (CRC Press, 2016).
- [24] W. Magnus, Commun. Pure Appl. Math. **7**, 649 (1954).
- [25] S. Blanes, F. Casas, J. A. Oteo and J. Ros, Phys. Rep. **470**, 151 (2009).
- [26] For a pedagogical presentation of the Magnus expansion see S. Blanes, F. Casas, J. A. Oteo and J. Ros, Eur. J. Phys. **31**, 907 (2010).
- [27] J. C. D’Olivo and J. A. Oteo, Phys. Rev. D **42**, 256 (1990); **54**, 1187 (1996).
- [28] J. C. D’Olivo, Phys. Rev. D **45**, 924 (1992); A. D. Supanitsky, J. C. D’Olivo, and G. A. Medina-Tanco, Phys. Rev. D **78**, 045024 (2008); A. N. Ioannisian and A. Yu. Smirnov, Nucl. Phys. B **816**, 94 (2009).
- [29] J. Fan and P. Langacker, JHEP **1204**, 083 (2012).
- [30] S. M. Bilenky, J. Hosek, and S.T. Petcov, Phys. Lett. B **94**, 495 (1980).
- [31] A. S. Dighe and A. Yu. Smirnov, Phys. Rev. D **62**, 033007 (2000).
- [32] J. N. Bahcall, *Neutrino Astrophysics* (Cambridge University Press, Cambridge, England, 1989).
- [33] G. L. Fogli, E. Lisi, D. Montanino, and A. Palazzo, Phys. Rev. D **65**, 073008 (2002); *ibid.* **66**, 039901(e) (2002).
- [34] A. M. Dziewonski and D. L. Anderson, Phys. Earth Planet. Inter. **25**, 297 (1981).
- [35] E. Hairer, S.P. Nørsett and G. Wanner, *Solving Ordinary Differential Equations I. Nonstiff Problems*, 2nd Ed. (Springer, Berlin, 1993).
- [36] J.M. Sanz-Serna and M.P. Calvo, *Numerical Hamiltonian Problems* (Chapman & Hall, London, 1994).
- [37] J.R. Dormand and P.J. Prince, J. Comput. Appl. Math. **6**, 19 (1980).
- [38] www.unige.ch/~hairer/prog/nonstiff/dopri5.f
- [39] F. Casas and A. Iserles, J. Phys. A: Math. Gen. **39**, 5445 (2006).
- [40] A. Iserles, BIT **42**, 561 (2002).
- [41] C. Moler and C. Van Loan, SIAM Review **45**, 3 (2003).
- [42] E. J. Putzer, Amer. Math. Monthly **73**, 2 (1966).

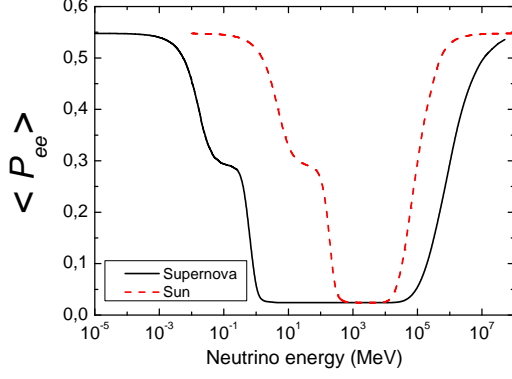


FIG. 1: Survival probability for electron neutrinos, Eq. (13), at the edge of the Sun (dashed line) and a supernova (solid line) as a function of the neutrino energy E . The curves have been obtained with the Magnus solver M4.

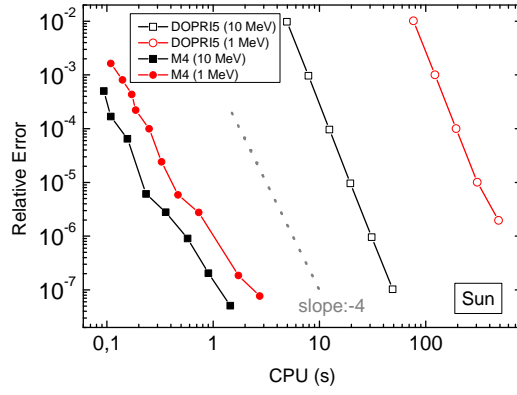


FIG. 2: Relative error of the solution of Eq. (27) as a function of the CPU cost for solar neutrinos with energies $E = 1$ MeV (circles) and $E = 10$ MeV (squares). Numerical integrations were carried out with the solvers M4 with variable stepsize (solid) and DOPRI5 (open).

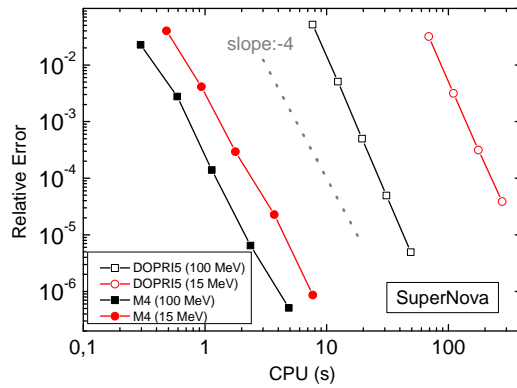


FIG. 3: Relative error of the solution of Eq. (27) as a function of the CPU cost for supernova neutrinos with energies $E = 15$ MeV (circles) and $E = 100$ MeV (squares). Numerical